



LENGUAJES DE PROGRAMACIÓN

Especialista en contenido:	LCDA. GRETTY LEAL	AUTORIZADO POR VICE RECTORADO ACADÉMICO (FIRMA Y SELLO)
Fecha de elaboración:	AGOSTO, 1992	
Elaborado por:	LCDA. GRETTY LEAL	

DESCRIPCIÓN DEL PROGRAMA

El programa cumple ciertos fundamentos teóricos de lenguaje de programación de alto nivel y sobre el diseño de los lenguajes de programación, definiciones, teoría formal y funcionamiento de compiladores; además una introducción a la programación de los lenguajes de alto nivel (Cobol, Fortran y Lenguaje C).

MODALIDAD Y ESTRATEGIA DE ENSEÑANZA

Conforme a la modalidad del curso se sugiere al docente:

- Revisión de literatura
- Explicación Teórico-práctica
- Ejercicios
- Asignación de trabajos

ESTRATEGIA DE ESTUDIO

Se sugiere una sólida preparación y revisión de los temas que serán impartidos. Además de la tarea individual, se recomienda formar equipos de trabajo para intercambiar opiniones y reforzar las destrezas conceptuales y prácticas.
Dar una máxima importancia a la solución de ejercicios en clase.

OBJETIVO TERMINAL DE LA ENSEÑANZA

Planteada una premisa, el estudiante habrá desarrollado la habilidad de seleccionar en forma apropiada el uso del lenguaje de programación más práctico para la aplicación requerida.

UNIDAD I	OBJETIVO TERMINAL DE UNIDAD
INTRODUCCIÓN A LOS LENGUAJES DE PROGRAMACIÓN	AL FINALIZAR LA UNIDAD, EL ESTUDIANTE SERÁ CAPAZ DE DIFERENCIAR EL CONCEPTO DE LENGUAJES NATURALES Y ARTIFICIALES, IDENTIFICAR LOS DIFERENTES TIPOS DE TRADUCTORES.
DURACIÓN	
1 SEMANA	
EVALUACIÓN	
ESTRATEGIA DE INSTRUCCIÓN	
Revisión de Bibliografía – Explicación Teórica - Práctica	
OBJETIVOS ESPECÍFICOS	
- EXPRESAR un recuento de los lenguajes de programación y evolución - EXPRESAR claramente qué son lenguajes de programación - IDENTIFICAR tipos de traductores y sus características específicas - DISCRIMINAR entre lenguajes naturales y artificiales	
CONTENIDOS	
1.1. Historia de los lenguajes de programación 1.2. Definición de lenguaje de programación 1.3. Definición y diferencia entre los tipos de traductores: Ensambladores, interpretadores y compiladores. 1.4. Definición y propósito de los lenguajes naturales y artificiales 1.5. Diferencia entre lenguajes naturales y artificiales 1.6. Sintaxis y semántica de los lenguajes naturales y artificiales	

UNIDAD II	OBJETIVO TERMINAL DE UNIDAD
LOS COMPILADORES	AL FINALIZAR LA UNIDAD, EL ESTUDIANTE SERÁ CAPAZ DE IDENTIFICAR LOS PROBLEMAS BÁSICOS QUE RESUELVE TODO COMPILADOR, COMO TAMBIÉN DISTINGUIR LA COMPLEJIDAD DE LOS LENGUAJES COMPILADOS.
DURACIÓN	
1 SEMANA	
EVALUACIÓN	
ESTRATEGIA DE INSTRUCCIÓN	
Revisión de Bibliografía, Explicación Teórico-práctica	
OBJETIVOS ESPECÍFICOS	
- EXPRESAR una idea clara de los compiladores - IDENTIFICAR estructuras y fases de un compilador - IDENTIFICAR los problemas básicos que debe resolver un compilador - EXPRESAR y distinguir la complejidad de los programas compiladores según la flexibilidad del lenguaje	
CONTENIDOS	
1.1. Problemas resueltos por medio del compilador 1.2. Análisis: Tabla de símbolos y literales 1.3. Análisis: Sintáctico en interpretación 1.4. Almacenamiento estático y dinámico 1.5. Dirección relativa y absoluta 1.6. Generación de código 1.7. Optimización independiente y dependiente de la máquina	

UNIDAD III	OBJETIVO TERMINAL DE UNIDAD
TEORÍA DE LOS LENGUAJES DE PROGRAMACIÓN	AL FINALIZAR LA UNIDAD, EL ESTUDIANTE SERÁ CAPAZ DE DISTINGUIR LOS ELEMENTOS QUE CONFORMAN UN LENGUAJE. IDENTIFICAR LAS ESPECIFICACIONES FORMALES DE LA SINTAXIS DE UN LENGUAJE
DURACIÓN	
1 SEMANA	
EVALUACIÓN	
ESTRATEGIA DE INSTRUCCIÓN	
Revisión Bibliográfica. Explicación Teórico-práctica	
OBJETIVOS ESPECÍFICOS	
- IDENTIFICAR todos los elementos que conforman un lenguaje de programación, señalándolos por escrito. - EXPRESAR claramente los conceptos de sistemas formales. - IDENTIFICAR especificaciones formales de un lenguaje de programación	
CONTENIDOS	
1.1. Elementos del lenguaje: Delimitadores, juego de caracteres y señales 1.2. Sistemas formales, símbolos terminales y no terminales. Diferentes sentencias. 1.3. Especificaciones de un lenguaje.	

UNIDAD IV		OBJETIVO TERMINAL DE UNIDAD
ESTRUCTURA DE LOS LENGUAJES DE PROGRAMACIÓN		AL FINALIZAR LA UNIDAD, EL ESTUDIANTE SERÁ CAPAZ DE DEMOSTRAR HABILIDAD EN LA IDENTIFICACIÓN DE ESTRUCTURAS BÁSICAS DEL LENGUAJE DE PROGRAMACIÓN.
DURACIÓN		
3 SEMANAS		
EVALUACIÓN		
ESTRATEGIA DE INSTRUCCIÓN		
Revisión de Bibliografía. Explicación Teórico-práctica. Ejercicios de Desempeño. Asignación de Trabajo.		
OBJETIVOS ESPECÍFICOS		
- Presentada una serie de ejemplos de programas en Cobol, Fortran, Lenguaje C; IDENTIFICAR los símbolos básicos para cada caso. - IDENTIFICAR elementos básicos. - Dada una serie de ejemplos de programas, RECONOCER e IDENTIFICAR expresiones, proposiciones y procedimientos.		
CONTENIDOS		
1.1. Conceptos básicos de programación en Cobol, Fortran y Lenguaje C. 1.2. Símbolos básicos para cada caso: Alfabeto, juego de caracteres, palabras reservadas, etc. 1.3. Elementos básicos para cada caso: Valores, identificadores, etiquetas, cadenas, etc. 1.4. Justificación y manipulación de expresiones y proposiciones. 1.5. Justificación, definición y activación de procedimientos para cada caso.		

UNIDAD V	OBJETIVO TERMINAL DE UNIDAD
SINTAXIS DE LAS INSTRUCCIONES EN COBOL, FORTRAN Y LENGUAJE C.	AL FINALIZAR LA UNIDAD, EL ESTUDIANTE SERÁ CAPAZ DE SELECCIONAR UN LENGUAJE DE PROGRAMACIÓN MÁS APROPIADO SEGÚN LA APLICACIÓN. PARA ELLO IDENTIFICARÁ LAS ESTRUCTURAS SINTÁCTICAS Y SUS REPRESENTACIONES.
DURACIÓN	
11 SEMANAS	
EVALUACIÓN	
ESTRATEGIA DE INSTRUCCIÓN	
Revisión de Bibliografía. Explicación Teórico-práctica. Ejercicios de Desempeño. Asignación de Trabajo.	
OBJETIVOS ESPECÍFICOS	
1. IDENTIFICAR las estructuras sintácticas de cada uno de los lenguajes (Cobol, Fortran, Lenguaje C). 2. SELECCIONAR estructuras de datos adecuados. 3. IDENTIFICAR estructuras lógicas de las instrucciones 4. DISTINGUIR estructuras lógicas y bloques de lenguaje	
CONTENIDOS	
1.1. Estructura de los lenguajes Cobol, Fortran, Lenguaje C. 1.2. Elementos de datos variables y declaraciones. 1.3. Estructuras de datos y memoria. 1.4. Expresiones simples y complejas. 1.5. Expresiones condicionales. 1.6. Estructuras de Control, Goto, Case... 1.7. Procedimientos: Activación, Invocación. 1.8. Instrucciones de entrada y salida.	

BIBLIOGRAFÍA

Alfred V. Aho y Jeffrey Ulmman. " Principles of Computer Desing".

B. With. "Systematic Programing"(Englewood Cliffs, N.J. Prentice-Hall, Inc.

J.E. Nicholls, "The Structure and Desing of Programming", Editor Addison Wesley (Idioma inglés).

Philippakis, "Cobol Structurado". McGraw-Hill